IN THE CLAIMS

1. (Original) A method comprising:

filling a cache line;

receiving a first request for a first segment of the cache line;

indicating at least the first segment is in a non-volatile state; and

sending at least the first segment while maintaining the cache line in one of a modified volatile state and an exclusive volatile state.

2. (Original) The method of claim 1, further comprising:

modifying at least a portion the first segment of the cache line; and

sending a notification of the modification.

3. (Original) The method of claim 1, further comprising:

modifying a second segment of the cache line without generating a notification of the modification; and

indicating the second segment is in a volatile state.

4. (Original) The method of claim 1, wherein the cache line is a part of a first cache associated with a first processor.

5. (Original) The method of claim 4, further comprising:

sending data from the cache line to a second cache associated with a second processor.

6. (Original) The method of claim 3, further comprising:

receiving a second request for a different third segment of the cache line; and

sending at least the third segment of the cache line while maintaining one of the modified volatile state and exclusive volatile state.

7. (Original) The method of claim 6, further comprising:

updating the cache line to indicate the third segment of the cache line is in a non-volatile state.

8. (Original) The method of claim 6, further comprising:

updating the cache line such that only the third segment of the cache line is in a non-volatile state; and

invalidating the cache line from all other processors holding the cache line or sending an updated copy of the cache line to a processor.

9. (Currently Amended) A memory device comprising:

a first plurality of memory segments to track a volatile status for a ~~subset~~ second plurality of a memory segments, wherein the volatile status comprises a modified volatile status, a shared volatile status, or an exclusive volatile status; and

circuitry to allow access to the plurality of memory segments.

10. (Original) The device of claim 9, wherein the volatile status is a modified volatile status.

11. (Original) The device of claim 9, wherein the volatile status is a shared volatile status.

12. (Original) The device of claim 9, wherein the volatile status is an exclusive volatile status.

13. (Currently Amended) A method comprising:

executing a first volatile load request;

placing requested data in a segment of a cache line; and

placing an indication of a shared volatile state associated with the requested data in the segment of the cache line.

14. (Original) The method of claim 13, further comprising:

executing a load or a second volatile load request for data held in the cache line in a non-volatile state; and

returning the result of the volatile load request.

15. (Original) The method of claim 13, further comprising:

executing a load or second volatile load request for a volatile portion of the cache line and placing the cache line in an invalid state.

16. (Original) The method of claim 13, further comprising:

executing a load or second volatile load request for a volatile portion of the cache line and receiving an updated copy of the cache line in a shared volatile state with requested data in a non-volatile state.

17. (Currently Amended) An apparatus comprising:

means for storing data; and

means for tracking ~~one of~~ a shared volatile state, a modified volatile state and an exclusive volatile state for the means for storing data.

18. (Original) The apparatus of claim 17, further comprising:

means for indicating one of a first portion and a second portion of a segment of the means for storing data contains non-volatile data.

19. (Original) The apparatus of claim 17, further comprising:

means for notifying a second means for storing data that a non-volatile data has been modified.

20. (Original) The apparatus of claim 17, further comprising:

means for indicating multiple segments are in one of a volatile and non-volatile state for a line of the means for storing data.

21. (Original) A system for enabling volatile shared data across caches comprising:

a first cache in a first central processing unit to store a first cache line in one of a shared volatile state, exclusive volatile state, a modified volatile state; and

a second cache in a second central processing unit in communication via a system interconnect with the first cache to store a second cache line.

22. (Original) The system of claim 21, further comprising:

a first processor associated with the first cache; and

a second processor associated with the second cache.

23. (Original) The system of claim 21, further comprising:

a system memory that is cached by the first and second caches.

24. (Original) The system of claim 21, wherein the first cache line indicates at least one non-volatile segment.

25. (Original) The system of claim 21, wherein the first cache notifies the second cache of a change in the non-volatile portion of a cache line in one of the modified volatile, the exclusive volatile state, and shared volatile state.

26. (Currently Amended) A processor comprising:

a pipeline to process instructions in one of program order and out of program order;

a set of execution units to execute the instructions; and

a set of caches coupled to the pipeline to store data required by the pipeline in ~~one of~~ a modified volatile, exclusive volatile, and shared volatile state.

27. (Original) The processor of claim 26, wherein the cache generates a notification upon modification of non-volatile data.

28. (Original) The processor of claim 26, wherein the cache shares data containing a modified portion.

29. (Original) A machine readable medium having instruction stored therein which when executed cause a machine to perform a set of operations comprising:

placing data in a cache line;

indicating the data in the cache line is in one of a modified volatile, exclusive volatile, and shared volatile state; and

sharing the data in the cache line.

30. (Original) The machine readable medium of claim 29, having instructions stored therein which when executed cause a machine to perform a set of operations further comprising:

generating a notification when a non-volatile data portion is modified.

31. (Original) The machine readable medium of claim 29, having instruction stored therein which when executed cause a machine to perform a set of operations further comprising:

indicating the size and position of a non-volatile portion of a cache line.

32. (New) The method of claim 1, wherein:

a non-volatile state requires that a modification to a segment of a cache line cause a notification of the modification to be sent; and

a volatile state requires that a modification to a segment of a cache line does not cause a notification of the modification to be sent.

33. (New) The method of claim 32, wherein:

a modified volatile state identifies a cache line having a non-volatile segment that is coherent between a plurality of caches associated with different processors, and a volatile segment that is not coherent between the plurality of caches;

an exclusive volatile state identifies a cache line having a non-volatile segment, a volatile segment, and a segment that is owned by a processor other than a processor associated with the cache; and

a shared volatile state identifies a cache line having a non-volatile segment that is shared between a plurality of caches associated with different processors, and a volatile segment that is shared between the plurality of caches.

34. (New) The method of claim 32, wherein the notification is sent to a processor that: does not own the modified segment, holds the modified segment in a cache line of a cache associated with the notified processor, or does not hold the modified segment in a cache line of a cache associated with the notified processor.

35. (New) The method of claim 32, wherein the cache line further comprises:

a lock field, a data field, and a status field, the status field to indicate a volatile status comprising one of a modified volatile state, a shared volatile state, and an exclusive volatile state.

36. (New) The method of claim 32, wherein the cache line further comprises a second segment in a volatile state and a third segment in a non-volatile state.

37. (New) The memory device of claim 9, wherein:

a modified volatile status identifies a cache line having a non-volatile segment that is coherent between a plurality of caches associated with different processors, and a volatile segment that is not coherent between the plurality of caches;

an exclusive volatile status identifies a cache line having a non-volatile segment, a volatile segment, and a segment that is owned by a processor other than a processor associated with the cache; and

a shared volatile status identifies a cache line having a non-volatile segment that is shared between a plurality of caches associated with different processors, and a volatile segment that is shared between the plurality of caches.

38. (New) The method of claim 13, wherein:

a modified volatile state identifies a cache line having a non-volatile segment that is coherent between a plurality of caches associated with different processors, and a volatile segment that is not coherent between the plurality of caches;

an exclusive volatile state identifies a cache line having a non-volatile segment, a volatile segment, and a segment that is owned by a processor other than a processor associated with the cache; and

a shared volatile state identifies a cache line having a non-volatile segment that is shared between a plurality of caches associated with different processors, and a volatile segment that is shared between the plurality of caches.

39. (New) The method of claim 38, wherein the cache line further comprises:

a lock field, a data field, and a status field, the status field to indicate a volatile status comprising one of a modified volatile state, a shared volatile state, and an exclusive volatile state.

40. (New) The method of claim 38, wherein the segment is a first segment; and wherein the cache line further comprises a second segment in a volatile state and a third segment in a non-volatile state, a non-volatile state requires that a modification to a segment of a cache line cause a notification of the modification to be sent, and a volatile state requires that a modification to a segment of a cache line does not cause a notification of the modification to be sent.

41. (New) The apparatus of claim 17, wherein:

a modified volatile state identifies a cache line having a non-volatile segment that is coherent between a plurality of caches associated with different processors, and a volatile segment that is not coherent between the plurality of caches;

an exclusive volatile state identifies a cache line having a non-volatile segment, a volatile segment, and a segment that is owned by a processor other than a processor associated with the cache; and

a shared volatile state identifies a cache line having a non-volatile segment that is shared between a plurality of caches associated with different processors, and a volatile segment that is shared between the plurality of caches.

42. (New) The apparatus of claim 41, further comprising:

means for tracking a lock field and a data field for the means for storing data.

43. (New) The apparatus of claim 41, further comprising:

means for tracking a volatile state and a non-volatile state for the means for storing data, a non-volatile state requires that a modification to a segment of a cache line cause a notification of the modification to be sent, and a volatile state requires that a modification to a segment of a cache line does not cause a notification of the modification to be sent.

44. (New) The system of claim 21, wherein:

a modified volatile state identifies a cache line having a non-volatile segment that is coherent between a plurality of caches associated with different processors, and a volatile segment that is not coherent between the plurality of caches;

an exclusive volatile state identifies a cache line having a non-volatile segment, a volatile segment, and a segment that is owned by a processor other than a processor associated with the cache; and

a shared volatile state identifies a cache line having a non-volatile segment that is shared between a plurality of caches associated with different processors, and a volatile segment that is shared between the plurality of caches.

45. (New) The processor of claim 26, wherein:

a modified volatile state identifies a cache line having a non-volatile segment that is coherent between a plurality of caches associated with different processors, and a volatile segment that is not coherent between the plurality of caches;

an exclusive volatile state identifies a cache line having a non-volatile segment, a volatile segment, and a segment that is owned by a processor other than a processor associated with the cache; and

a shared volatile state identifies a cache line having a non-volatile segment that is shared between a plurality of caches associated with different processors, and a volatile segment that is shared between the plurality of caches.

46. (New) The machine readable medium of claim 29, wherein:

a modified volatile state identifies a cache line having a non-volatile segment that is coherent between a plurality of caches associated with different processors, and a volatile segment that is not coherent between the plurality of caches;

an exclusive volatile state identifies a cache line having a non-volatile segment, a volatile segment, and a segment that is owned by a processor other than a processor associated with the cache; and

a shared volatile state identifies a cache line having a non-volatile segment that is shared between a plurality of caches associated with different processors, and a volatile segment that is shared between the plurality of caches.